আন্তর্জাতিক ইসলামী বিশ্ববিদ্যালয় চট্টগ্রাম
الجامعة الإسلامية العالمية شيتاغونغ
International Islamic University Chittagong

CSE – 3532 Assignment

Course code: CSE-3532

Course Title: Tools and Technologies for Internet Programming.

Submission Date: 07/ 08 / 2023

Submitted by:

Name: Motafa Zamal Zahid

ID: C213035

Section: 4AM

Department: Computer Science and Engineering

Submitted to:

Mohammad Shabaj Khan

Adjunct Faculty,

Computer Science and Engineering

International Islamic University Chittagong

Ans. to the Q. No. (a)-

Block-level elements: Block-level elements begin on a new line and occupy the full available width within their parent container. They create a "block" that pushes other elements horizontally. Block-level elements can encompass other block-level and inline elements. Examples of block -level elements include:

-- <section>: Represents a thematic grouping of content.

-- <article>: Represents a self-contained piece of content.

-- <header>: Represents a group of introductory content at the top of a section or page.

-- <nav>: Represents navigation links.

-- <main>: Represents the main content of the document.

-- <footer>: Represents the footer section of a document or a section.

Inline elements: Inline elements do not initiate a new line and only occupy the necessary width to fit their content. They flow within the text and do not disrupt the surrounding content. Inline elements cannot contain block-level elements but can include other inline elements. Examples of Inline elements include:

-- : A generic inline container used for styling and scripting purposes.

-- <a>: Represents a hyperlink.

-- : Used for indicating strong importance.

-- : Used for emphasizing text.

-- <img>: Displays an image.

-- <input>: Used for input fields in forms.

-- <button>: Represents a clickable button.

In summary, block-level elements create blocks that occupy the entire width, while inline elements flow within the text and take up only the

necessary width. Block-level elements can encompass both block-level and inline elements, but inline elements cannot contain block-level elements.

Ans. to the Q. No. (b)-

Semantic tags in HTML are elements specifically designed to carry explicit meaning and convey the structural organization of the enclosed content. They offer valuable context and define the purpose of the content, making it easier for automated systems (such as search engines) and developers to understand the document's layout. Semantic tags play a crucial role in enhancing the accessibility, search engine optimization (SEO), and maintainability of web pages by organizing the content in a meaningful and logical manner.

Examples of Semantic Tags:

-- <header>: Represents the introductory section, often containing logos, headings, and navigation menus.

-- <nav>: Defines a section that contains navigation links, guiding users to various parts of the website.

-- <main>: Represents the primary content of the web page, excluding headers, footers, and sidebars.

-- <article>: Represents an independent, complete, and self-contained piece of content, such as a blog post or news article.

-- <section>: Represents a thematic grouping of content within the document.

Examples of Non-Semantic Tags:

-- : A generic container without any inherent meaning or semantic value, often used for layout purposes.

-- : A generic inline container utilized for applying styles or scripting purposes, lacking any specific meaning.

-- <b>: Represents bold text styling but does not convey the reason for

the text's importance or significance.

-- <i>: Represents italicized text but does not provide context for why the text is in italics.

-- <font>: Used to apply font styles, sizes, and colors but lacks semantic meaning and is not recommended for modern web development.

In essence, semantic tags enrich the structure and semantics of the HTML document, making it more accessible, SEO-friendly, and easier to maintain. They effectively convey the purpose and relationship of content, fostering better understanding for both humans and machines. By using semantic tags, developers can create more meaningful and well-organized web pages, ultimately leading to better user experiences and improved search engine rankings.

Ans. to the Q. No. (c)-

Ordered list: A numbered list is utilized to present a series of items in a specific sequence, where each item is preceded by a number or letter (typically starting from 1). The list items' order holds significance and can be indicated using numerical or alphabetical ordering.

Ordered Lists (<ol>): - Used to generate a list with numbers or letters as markers. - Each list item is prefixed with a number (default is sequential numbering). - The order of items in the list matters, and they follow a specific sequence.

Unordered list: An unordered list is employed to represent a collection of items with no defined sequence or order. Each item in the list is preceded by a bullet point (customizable using CSS) or other marker.

Unordered Lists (<ul>): - Utilized to create a list with bullet points or other custom markers. - Each list item is preceded by a bullet point (default) or any customized marker. - The order of items in the list does not carry significance; they form a simple collection of points.

In summary, an ordered list is used to present items in a specific sequence with numerical or alphabetical markers, while an unordered list is employed to display a collection of items with no specific order, using bullet points or custom markers. Both types of lists serve distinct

purposes in structuring and organizing content on web pages.

Ans. to the Q. No. (d)-

inline

The inline styles will exclusively impact the HTML element to which the style attribute with CSS-property values is implemented. The initial paragraph in the example below will be tailored with a red color and a font size of 20px. The properties exclusively affect the initial line of the code, not the entire code.

Example :

```
<body>
<p style="color:blue; font-size: 10px;">This is our first HTML code.</p>
        <p>This is our second HTML code</p>
</body>
```

internal

Embedded CSS is one of the most favored CSS approaches for modifying, customizing, and adjusting the distinctive styles of an individual web page. You can employ embedded CSS by incorporating the <style> element within the <head> section of an HTML web page. Embedded CSS is suitable for styling a single web page, although it does not apply to multiple web pages. However, you can utilize the same code to style multiple web pages.

Example :

```
<!DOCTYPE html>
<html>
<head>
 <title>Internal Stylesheet Example</title>
 <style>
   body {
     font-family: Arial, sans-serif;
     background-color: #f0f0f0;
   }
   h1 {
     color: #007bff;
   }
 </style>
</head>
```

```
<body>
  <!-- Your HTML content goes here -->
</body>
</html>
```

External CSS:

In this method, you link an external CSS file to your HTML document using the <link> element within the <head> section of the HTML file.

Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>Inline Styles Example</title>
</head>
<body>
  <h1 style="color: #eee;">This is a heading with inline style.</h1>
  <p style="font-size: 10px; color: #fff;">This is a paragraph with inline style.</p>
</body>
</html>
```

Ans. to the Q. No. (e)-

Content Width: The content is explicitly set to 300px.

Border Width: The overall border size is 30px (15px on the left + 15px on the right) since the border has a width of 15px on each side.

Padding Width: The combined padding measures 100px (50px on the left + 50px on the right) as the padding is 50px wide on each side.

Margin Width: The total margin span is 40px (20px on the left + 20px on the right) since the margin extends 20px on each side.

To calculate the total width of the element:

Total width = Content width + Total Padding width + Total Border width + Total Margin width

Total width = 300px + 200px + 30px + 40px = 570px

Therefore, the element will have a total width of 570px.

Ans. to the Q. No. (f)-

A pseudo-class is a selector that allows you to target elements in specific conditions, such as being the first element of its kind or being interacted with by the user, like hovering over it with the mouse pointer. These selectors work as if you applied a class to a certain part of your document, which can reduce the need for excessive classes in your code and make it more adaptable and maintainable.

Styling Based on State: Pseudo-classes enable you to style elements based on their current state or user interactions. For instance, you can change the appearance of a link when it's hovered over or clicked, modify the style of a form element when it's in focus, or apply styles to an element when it's selected.

Styling Based on Position: Pseudo-classes also allow you to target elements based on their position within the document tree. For example, you can style the first child or last child of a parent element differently, select even or odd elements, or target specific elements based on their relationship with other elements.

Creating Interactive Effects: Pseudo-classes are essential for generating interactive and dynamic effects on web pages. They make it easier to create animations, transitions, and other effects based on user interactions or element states.

Ans. to the Q. No. (g)-

The CSS rule margin: 15px 70px; adds space around an element, creating a gap between it and nearby elements.

In this rule, the margin property is used with two values: 15px and 70px. These values correspond to the top/bottom margin and left/right margin, respectively. The order of the values is crucial and follows the shorthand notation for specifying margins.

To elaborate on the values:

--15px: This value represents the vertical margins, adding a margin of 15 pixels above and below the element.

--70px: This value signifies the horizontal margins, providing a margin of 70 pixels on the left and right sides of the element.

As a result, the CSS rule margin: 15px 70px; will establish a margin of 15 pixels above and below the element and a margin of 70 pixels on the left and right sides of the element.

This technique is commonly used to create spacing between an element and its surrounding

content or other elements on the page. You can adjust the specific values as needed to achieve the desired layout and visual appearance.

Ans. to the Q. No. (h)-

The CSS descendant selector is a valuable tool used to target elements that are children of a specific parent element. In this context, "descendant" refers to elements nested anywhere within the DOM tree, whether they are direct children or located deeper within multiple levels of nesting.

The descendant combinator is denoted by a single space between two selectors. It combines two selectors: the first selector represents the ancestor (parent, grandparent, etc.), and the second selector represents the descendants. Elements matched by the second selector will be selected if they have a parent element that matches the first selector. The descendant selector employs the concept of descendant combinators to achieve this type of selection.

In essence, the CSS descendant selector allows you to select and style elements based on their hierarchical relationship with another specific element in the HTML document. It provides a powerful and flexible way to apply styles to elements within a specific context and hierarchy on the page.